AMENDMENT TO CLAIMS:

1. (currently amended) A system for memory management, the system comprising:

a tag controlled buffer in communication with a memory device, said memory device including a plurality of pages divided into a plurality of individually addressable lines, wherein said tag controlled buffer includes:

a prefetch buffer including at least one of the individually addressable lines from the memory device; and

a tag cache in communication with the prefetch buffer, the tag cache including at least one ~~plurality of~~ tag~~s~~, wherein each said tag is associated with one of the pages in the memory device, each said tag includes a reference history field and a pointer to at least one ~~of the~~ line~~s~~ in the prefetch buffer that is from the associated page, ~~and~~ access to the lines in the prefetch buffer is controlled by the tag cache, and the reference history field includes information about how the lines from the associated page have been accessed in the past and is utilized to determine which lines in the associated page should be added to the prefetch buffer when the tag is added to the tag cache.

2. (original) The system of claim 1 wherein access includes at least one of inserting new lines into the prefetch buffer, deleting one of the lines from the prefetch buffer and reading one of the lines in the prefetch buffer.

3. (original) The system of claim 1 wherein the tag controlled buffer transmits one of the lines in the prefetch buffer to a cache device in response to a command from the tag cache.

4. (original) The system of claim 1 wherein the prefetch buffer is implemented by a random access memory.

5. (original) The system of claim 1 wherein all of the lines in the prefetch buffer corresponding to one of the tags in the tag cache are deleted in response to the tag being deleted from the tag cache.

6. (original) The system of claim 1 wherein the system further comprises instructions to implement a replacement algorithm for the tags in the tag cache and for the lines in the prefetch buffer, wherein upon insertion of a new tag into the tag cache, a sufficient number of the tags are removed to make space for the new tag and for the prefetch lines associated with the new tag.

7. (original) The system of claim 6 wherein the tags that are removed are written back to a lower level memory.

8. (currently amended) The system of claim 6 wherein the tags that are removed are written back to a lower level memory and the tags in the lower level memory include a page identifier field and the~~a~~ reference history field.

9. (original) The system of claim 8 wherein the tags in the lower level memory further include a flags field.

10. (original) The system of claim 1 wherein the system further comprises instructions to implement a method comprising updating the tags in the tag cache in response to the lines in the prefetch buffer being inserted, deleted, modified and referenced.

11. (original) The system of claim 1 wherein the system further includes instructions to implement a method comprising:

receiving a fault notification from a cache device, the fault notification including a fault page identifier and a fault line identifier;

accessing the tag cache to determine if one of the lines in the prefetch buffer corresponds to the fault page identifier and the fault line identifier;

transmitting the line corresponding to the fault page identifier and the fault line identifier from the prefetch buffer to the cache device in response to locating the line corresponding to the fault page identifier and the fault line identifier in the prefetch buffer; and

updating the tag cache to reflect the transmitting.

12. (currently amended) The system of claim 1 wherein each tag in the tag cache further includes a page identifier field~~,~~ and a presence bits field.~~, a history vector field, and a pointers field.~~

13. (original) The system of claim 12 wherein each tag in the tag cache further includes a flags field.

14. (currently amended) A system for memory management, the system comprising:

a random access memory including at least one line, wherein each line is associated with a page in a memory device and space in the random access memory is allocated on per line basis; and

a first cache device including a plurality of tags, wherein each tag corresponds to one of the pages in the memory device,~~ and~~ each tag indicates the location in the random access memory of the at least one line associated with the page, and each tag includes a reference history field that includes information about how lines in the associated page in the memory device have been accessed in the past and is utilized to determine which lines from the associated page should be added to the random access memory when the tag is added to the first cache device.

15. (original) The system of claim 14 further comprising a computer processor, the computer processor including instructions to transmit one of the lines in the random access memory to a second cache device in response to a command from the first cache device.

16. (original) The system of claim 14 wherein all of the lines in the random access memory corresponding to one of the tags in the first cache device are deleted in response to the tag being deleted from the tag cache.

17. (original) The system of claim 14 further comprising a computer processor, the computer processor including instructions to implement a replacement algorithm for the tags in the first cache device and for the lines in the random access memory, wherein upon insertion of a new tag into the first cache device, a sufficient number of the tags are removed to make space for the new tag and for the prefetch lines associated with the new tag.

18. (original) The system of claim 17 wherein the tags that are removed are written back to a lower level memory.

19. (currently amended) The system of claim 17 wherein the tags that are removed are written back to a lower level memory and the tags in the lower level memory include a page identifier field and the~~a~~ reference history field.

20. (original) The system of claim 19 wherein the tags in the lower level memory further include a flags field.

21. (original) The system of claim 14 further comprising a computer processor, the computer processor including instructions to implement a method for updating the tags in the first cache device in response to the lines in the random access memory being inserted, deleted, modified and referenced.

22. (currently amended) The system of claim 14 wherein each tag in the first cache device further includes a page identifier field~~,~~ and a presence bits field.~~, a history vector field, and a pointers field.~~

23. (original) The system of claim 22 wherein each tag in the first cache device further includes a flags field.

24. (currently amended) A method for memory management, the method comprising:

receiving a fault notification from a first cache device, the fault notification including a fault page identifier and a fault line identifier;

accessing a second cache device to determine if a line in a random access memory corresponds to the fault page identifier and the fault line identifier, wherein:

the random access memory includes at least one line associated with a page in a memory device;

the second cache device includes a plurality of tags each corresponding to one of the pages in the memory device; ~~and~~

each tag indicates the location in the random access memory of the at least one line associated with the page corresponding to the tag; and

each tag includes a reference history field that includes information about how lines in the corresponding page have been accessed in the past and is utilized to determine which lines from the corresponding page should be added to the random access memory when the tag is added to the second cache device;

transmitting the line corresponding to the fault page identifier and the fault line identifier from the random access memory to the first cache device in response to the accessing resulting in locating the line corresponding to the fault page identifier and the fault line identifier in the random access memory; and

updating the tag in the second cache device corresponding to the fault page identifier to reflect the transmitting.

25. (currently amended) A method for memory management, the method comprising:

receiving a fault notification from a requestor, the fault notification including a fault page identifier and a fault line identifier;

determining if a tag corresponding to the fault page identifier is located in a tag cache, wherein the tag cache includes a plurality of tags, each tag includes a reference history field including information about how lines in the corresponding page have been accessed in the past and at least one pointer to lines in a prefetch buffer~~, and the tag cache designates at least one prefetch line~~;

in response to locating the tag corresponding to the fault page identifier:

transmitting a line corresponding to the fault line identifier to the requestor; and

updating the tag corresponding to the fault page identifier to reflect the transmitting; and

in response to not locating the tag corresponding to the fault page identifier:

inserting a new tag corresponding to the fault page identifier into the tag cache;

transmitting the line corresponding to the fault line identifier to the requestor; and

inserting~~ the~~ prefetch lines associated with~~included in~~ the new tag into the prefetch buffer, wherein the inserting is performed via the tag cache and the prefetch lines are determined in response to contents of the reference history field in the new tag.

26. (original) The method of claim 25 further comprising retrieving the line corresponding to the fault line identifier from a memory device.

27. (original) The method of claim 25 further comprising retrieving the line corresponding to the fault line identifier from the prefetch buffer, wherein the retrieving is via the tag cache.

28. (original) The method of claim 25 wherein the requestor is a cache device.

29. (currently amended) A computer program product for cache memory management, the computer program product comprising:

a storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method comprising:

receiving a fault notification from a requestor, the fault notification including a fault page identifier and a fault line identifier;

determining if a tag corresponding to the fault page identifier is located in a tag cache, wherein the tag cache includes a plurality of tags, each tag includes a reference history field including information about how lines in the corresponding page have been accessed in the past and at least one pointer to lines in a prefetch buffer~~, and the tag cache designates at lest one prefetch line~~;

in response to locating the tag corresponding to the fault page identifier:

transmitting a line corresponding to the fault line identifier to the requestor; and

updating the tag corresponding to the fault page identifier to reflect the transmitting; and

in response to not locating the tag corresponding to the fault page identifier:

inserting a new tag corresponding to the fault page identifier into the tag cache;

transmitting the line corresponding to the fault line identifier to the requestor; and

inserting~~ the~~ prefetch lines associated with~~included in~~ the new tag into the prefetch buffer, wherein the inserting is performed via the tag cache and the prefetch lines are determined in response to contents of the reference history field in the new tag.